

## Analisis Dan Perbandingan Metode *Sobel Edge Detection* Dan *Prewitt* Pada Deteksi Tepi Citra Daun Srilangka

Beriman Sitohang<sup>1\*</sup>, Anita Sindar<sup>1</sup>  
<sup>1,2</sup> STMIK Pelita Nusantara  
Jl. Iskandar Muda No. 1 Medan 20154

Corresponding author's e-mail: berimansitohang16@yahoo.com

**Abstrak**— Daun srilangka dikenal dengan nama daun sirsak, berbentuk bulat lonjong. Identifikasi citra berdasarkan tepi (outline) objek dalam Pengolahan citra menggunakan edge detection. Operasi pelacakan tepi untuk menemukan perubahan intensitas lokal yang berbeda dalam sebuah citra. Penerapan Sobel dan Prewitt adalah untuk mengurangi noise sebelum melakukan perhitungan deteksi tepi sehingga tepi-tepi yang dihasilkan lebih banyak. Deteksi Tepi Sobel dan prewitt dapat mendeteksi tepian yang sebenarnya dengan tingkat error yang minimum. Sobel dan Prewitt mampu mengidentifikasi tepi dengan sumbu-sumbu yang digunakan, memiliki intensitas bilangan acak yang hasilnya berupa angka 0 dan 255 secara menyebar. Sistem yang dibangun menampilkan konversi citra RGB, Grayscale, Sobel dan Prewitt dan Histogram dari proses citra yang dilakukan. Hasil analisa metode Sobel dan Prewitt ditampilkan dalam grafik histogram.

Kata kunci: Deteksi Tepi, Histogram, Noise, Prewitt, Sobel

**Abstract**— *Srilanka leaves are known as soursop leaves, which are oval in shape. Image identification based on the edges (outline) of objects in image processing using edge detection. Edge tracing operation to find different local intensity changes in an image. The application of Sobel and Prewitt is to reduce noise before performing edge detection calculations so that more edges are generated. Edge Detection Sobel and prewitt can detect true edges with minimum error rate. Sobel and Prewitt were able to identify the edges with the axes used, having random number intensities whose results are scattered numbers 0 and 255. The system built displays the conversion of RGB, Grayscale, Sobel and Prewitt images and histograms from the image process that is carried out. The analysis results of the Sobel and Prewitt methods are displayed in a histogram graph.*  
**Keyword:** *Edge Detection, Histogram, Noise, Prewitt, Sobel*

### 1. Pendahuluan

Tanaman Srilangka (Sirsak) adalah tanaman asli Indonesia. Tanaman Srilangka memiliki tinggi pohon sekitar 8 meter. Batang coklat berkayu, bulat, bercabang. Mempunyai daun berbentuk telur atau lanset, ujung runcing, tepi rata, pangkal meruncing, pertulangan menyirip, panjang tangkai 5 mm, hijau kekuningan. Daun Srilangka memiliki variasi dalam segi bentuk, warna dan ukuran daun. Struktur tulang daun sebagai salah satu fitur yang dapat membedakan daun dari berbagai jenis tumbuhan karena tulang daun memiliki fitur yang bermacam-macam bentuk pada setiap jenis tumbuhan. Metode Sobel dan Prewitt dapat digunakan untuk mengidentifikasi bentuk daun Srilangka. Setelah mendapatkan hasil gambar (citra), maka dilakukanlah proses untuk mengidentifikasi gambar (citra) daun Srilangka tersebut untuk dikenali karakteristik, tekstur, dan bentuk dari daun Srilangka tersebut.

Ada beberapa metode deteksi tepi yang umum sering digunakan untuk pendeteksian tepi yaitu Prewitt, Roberts, Sobel, Canny dan sebagainya [1]. Dalam hal ini, peneliti akan menggunakan metode sobel dan metode prewitt dalam pendeteksian tepi citra. Alasan peneliti memilih metode itu, karena kedua metode ini dapat menghasilkan kualitas citra yang lebih baik dalam pendeteksian tepi dibandingkan metode lainnya. Kelebihan dari metode Sobel dan Prewitt adalah kemampuan untuk mengurangi noise sebelum melakukan perhitungan deteksi tepi sehingga tepi-tepi yang dihasilkan lebih banyak [2]. Deteksi Tepi Sobel dan Prewitt dapat mendeteksi tepian yang sebenarnya dengan tingkat error yang minimum dengan kata lain, metode Sobel dan Prewitt didesain untuk menghasilkan citra tepian yang optimal [3]. Metode Sobel dan prewitt menempatkan penekanan atau pembobotan pada piksel-piksel yang lebih dekat dengan titik pusat jendela, sehingga pengaruh piksel-piksel tetangga akan berbeda sesuai dengan letaknya terhadap titik di mana gradien dihitung. Dari susunan nilai-nilai pembobotan pada jendela juga terlihat bahwa perhitungan terhadap gradien juga merupakan gabungan dari posisi mendatar dan posisi vertical [4]. Beberapa peneliti terdahulu yang sudah melakukan pendeteksi tepi, misalnya Yudiyanto dan Murinto melakukan penelitian dengan metode Canny, diperoleh hasil

dari deteksi tepi menggunakan metode Canny menghasilkan titik tepian yang bersambung cukup jelas sehingga mempermudah dalam uji mutu daun. Hastuti juga melakukan penelitian membandingkan metode Canny dengan beberapa metode, salah satunya membandingkan dengan metode Prewitt, dimana hasil deteksi tepi canny memperoleh deteksi tepi citra lebih baik dibandingkan metode Prewitt dan metode lainnya [5]. Peneliti sebelumnya juga membandingkan metode sobel dengan beberapa metode, salah satunya membandingkan dengan metode prewitt, dimana pengaruh komposisi warna terhadap deteksi tepi metode Sobel lebih baik dibandingkan dengan metode prewitt. Kedua metode ini menghasilkan citra yang berbeda, dimana kelebihan metode Sobel merupakan kemampuan untuk mengurangi noise sebelum melakukan perhitungan deteksi tepi [6]. Metode canny terkenal sebagai operator deteksi tepi yang optimal. Algoritma ini memberikan tingkat kesalahan rendah, melokalisasi titik-titik tepi (jarak piksel-piksel) yang ditemukan deteksi dan tepi yang sesungguhnya sangat pendek) dan hanya memberikan satu tanggapan untuk satu tepi [7]. Identifikasi masalah dalam penelitian ini mengidentifikasi tesktur, bentuk, ukuran dan keunikan dari citra daun Srilangka dengan menerapkan perbandingan metode Sobel dan Prewitt menggunakan bahasa pemrograman Matlab R2015a [8].

## 2. Tinjauan Pustaka

### 2.1. Derau (Noise)

Noise adalah suatu gangguan yang disebabkan oleh penyimpanan data digital yang diterima oleh alat penerima data gambar yang dapat mengganggu kualitas citra [14]. Noise dapat disebabkan oleh gangguan fisik (optik) pada alat penangkap citra misalnya kotoran debu yang menempel pada lensa foto maupun akibat proses pengolahan yang tidak sesuai [9].

### 2.2. Pendeteksian Tepi (Edge detection)

Deteksi tepi akan menghasilkan nilai tinggi apabila ditemukan tepi dan nilai rendah jika sebaliknya. Pelacakan tepi merupakan operasi untuk menemukan perubahan intensitas lokal yang berbeda dalam sebuah citra. Deteksi tepi merupakan salah satu operasi dasar dari pemrosesan citra [10]. Tepi merupakan batas dari suatu objek. Pada proses klasifikasi citra, deteksi tepi sangat diperlukan sebelum pemrosesan segmentasi citra. Batas objek suatu citra dapat dideteksi dari perbedaan tingkat keabuannya. Tepi atau sisi dari sebuah obyek adalah daerah di mana terdapat perubahan intensitas warna yang cukup tinggi. Proses deteksi tepi (edge detection) akan melakukan konversi terhadap daerah ini menjadi dua macam nilai yaitu intensitas warna rendah atau tinggi, contoh bernilai nol atau satu. Deteksi tepi akan menghasilkan nilai tinggi apabila ditemukan tepi dan nilai rendah jika sebaliknya [11]. Faktor kunci dalam mengekstraksi ciri adalah kemampuan mendeteksi keber-adaan tepi (edge) dari objek di dalam citra. Pendeteksian tepi merupakan langkah pertama untuk melingkupi informasi di dalam citra. Tepi mencirikan batas-batas objek dan karena itu tepi berguna untuk proses segmentasi dan identifikasi di dalam citra. Tujuan pendeteksian tepi adalah untuk meningkatkan penampakan garis batas suatu daerah atau objek di dalam citra. Untuk mendeteksi tepi-tepi pada citra ini dapat digunakan metode Sobel, Prewitt, Robert, Laplacian of a Gaussian, Canny, dan lain-lain. Operator Sobel adalah operator yang paling banyak digunakan sebagai pelacak tepi karena kesederhanaan dan keampuhannya [12]. Selain itu metode Sobel merupakan pengembangan metode Robert dengan menggunakan filter HPF yang diberi satu angka nol penyangga. Metode ini mengambil prinsip dari fungsi laplacian dan gaussian yang dikenal sebagai fungsi untuk membangkitkan HPF [13]. Kelebihan dari metode sobel ini adalah kemampuan untuk mengurangi noise sebelum melakukan perhitungan deteksi tepi. Pengembangan dari gradient operator dengan menggunakan 2 mas k (horizontal dan vertikal) ukuran 3x3. Pada operator ini kekuatan gradient ditinjau dari sudut pandang horizontal dan vertikal pada posisi horizontal dan vertikal). Selain itu metode Prewitt merupakan pengembangan metode Robert dengan menggunakan filter HPF yang diberi satu angka nol penyangga. Metode ini mengambil prinsip dari fungsi laplacian yang dikenal sebagai fungsi untuk membangkitkan HPF.

## 3. Metode Penelitian

Perancangan aplikasi perbandingan metode sobel dan prewit yang dibangun harus memiliki data-data yang akan diolah dan digunakan pada sistem yang nantinya untuk dilakukan untuk membandingkan kedua metode. Analisis kebutuhan sistem :

- Input : data citra *format jpg* atau *jpeg* sebagai bahan uji sistem yang akan dibangun.
- Process : melakukan analisis perbandingan metode sobel dan prewit beserta perhitungannya tentang Analisis dan Perbandingan Metode Sobel dan Prewit pada deteksi citra daun srilangka.
- Output : hasil analisis metode dan perhitungan berupa citra yang telah dibandingkan dengan metode sobel dan prewit.

Tahapan dalam melakukan metode Sobel dan Prewitt:

1. Lakukan proses identitas citra RGB.
2. Konversi citra RGB ke dalam tabel piksel
3. Hasil konversi citra rgb di konversi ke grayscale
4. Lakukan proses metode sobel dan prewit
5. Hasil akhir dalam bentuk histogram citra.

#### 4. Hasil dan Pembahasan

Data Citra yang digunakan yaitu citra RGB atau warna dengan identitas nama file Srilangka2.jpeg, ukuran : 36041, panjang : 493, lebar : 1040, 24 bit. Data Citra RGB (*Red, Green, Blue*) sebagai data uji memiliki piksel penyusun berupa angka yang disebut sebagai data piksel atau data matriks. Data Piksel dapat ditampilkan dengan menggunakan program Matlab sebanyak 10 x 10 piksel yang akan diuji prosesnya.

Tabel 1. Citra RGB

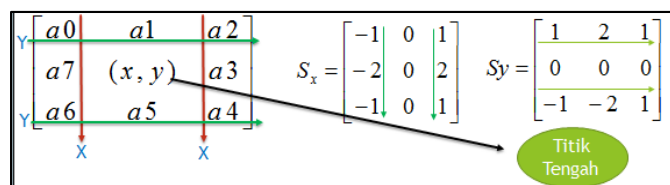
| Citra Asli  | Potongan Citra dengan Tools   |
|---|---|
|  |  |

Pemotongan citra untuk dijadikan sampel perhitungan sebesar 10 x 10 piksel citra pada prosesnya dimulai dari sumbu y sejauh 450 dan sumbu x sejauh 270, dengan nilai RGB (*Red, Green, Blue*) nya [127, 151, 115]. Masing-masing x dan y diambil sebanyak 10 piksel. Sehingga untuk y dimulai dari 450 sampai 459, dan untuk x dimulai dari 270 sampai 279. Pada citra RGB atau citra warna, terdiri dari komponen *Red, Green, dan Blue* dan masing-masing komponen memiliki nilai matriks. Citra RGB dan citra Grayscale memiliki perbedaan dalam informasi citra penyusunnya, Jumlah warna pada citra grey adalah 256, karena citra grey jumlah bitnya adalah 8, sehingga jumlah warnanya adalah  $2^8=256$ , nilainya berada pada jangkauan 0-255, sehingga nilai intensitas dari citra grey tidak akan melebihi 255 dan tidak kurang dari 0. Model penyimpanannya adalah  $f(x,y)$  = nilai intensitas, dengan x dan y merupakan posisi nilai intensitas. Citra uji diatas dengan ukuran lebar 720 dan tinggi 960, maka jumlah byte yang diperlukan untuk penyimpanan citranya:  $720 \times 960 \times 3 = 2073600$ . Sedangkan jumlah warna untuk citra RGB adalah dengan mengalikan jumlah pada masing-masing komponennya, jumlah dari tiap komponennya, R=255 (8bit). G=255 (8bit) dan B=255 (8bit) yang sering disebut sebagai citra dengan intensitas 24 bit atau *truecolor*. Jumlah warna RGB adalah sejumlah  $2^8 \times 2^8 \times 2^8 = 16.777.216$ . Sehingga jumlah byte yang diperlukan untuk file citra RGB lebih 3 kali space file *grayscale*. Setiap piksel dari citra *grayscale* 256 gradasi warna diwakili oleh 1 byte dan 1 piksel citra RGB diwakili oleh 3 byte, dimana masing-masing byte merepresentasikan warna *Red, Green, Blue* didalam memori. Citra RGB dapat diubah menjadi citra Grayscale dengan cara menghitung rata-rata elemen warna *Red, Green,*

*Blue*. Secara sistematis perhitungannya :  $f_0 = (x, y) = \frac{f_i^R(x, y) + f_i^G(x, y) + f_i^B(x, y)}{3}$

Keterangan: f sebagai variableR, G, B sebagai warna *Red, Green, Blue*; X,y sebagai sumbu titik.

Data piksel awal pada baris 1 kolom 1 (x,y)1,1= (196 +200+202)/3=199,33, maka dilakukan pembulatan menjadi 199 dan langkah tersebut dilanjutkan lagi sampai selesai. Peninjauan pengaturan pixel di sekitar pixelnya (x,y) adalah :



Gambar 1. Piksel Awal Citra

$F(x,y)$  = ukuran piksel citra dalam matriks;  $a_n$  = nilai piksel pada matriks ;(x,y) = nilai piksel baru hasil konvolusi ; Turunan parsial dihitung dengan: $S_x = (a_2+ca_3+a_4)-(a_0+ca_7+a_6) =$  Sumbu Vertikal ;  $S_y =$

$(a_0+ca_1+a_2)-(a_6+ca_5+a_4) = \text{Sumbu Horizontal}$ . G adalah besar gradient ditengah titik kernel.

Tabel 2. Konversi Citra

|     |     |     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 131 | 123 | 113 | 111 | 117 | 122 | 122 | 121 | 122 | 121 |
| 131 | 120 | 108 | 109 | 114 | 118 | 119 | 121 | 123 | 122 |
| 127 | 117 | 107 | 106 | 110 | 114 | 116 | 119 | 124 | 123 |
| 117 | 112 | 106 | 105 | 108 | 110 | 113 | 118 | 124 | 123 |
| 110 | 110 | 107 | 105 | 106 | 107 | 110 | 116 | 123 | 122 |
| 113 | 115 | 114 | 105 | 106 | 105 | 108 | 115 | 122 | 121 |
| 121 | 123 | 121 | 106 | 105 | 105 | 107 | 114 | 121 | 120 |
| 116 | 112 | 108 | 107 | 110 | 110 | 108 | 108 | 112 | 115 |
| 113 | 110 | 107 | 107 | 111 | 111 | 108 | 108 | 112 | 115 |
| 109 | 108 | 107 | 108 | 113 | 112 | 110 | 109 | 112 | 115 |

Dengan data citra hasil konversi dilakukanlah proses algoritma sobel sebanyak  $n$  iterasi:

Tabel 3. Iterasi 1

|     |     |     |
|-----|-----|-----|
| 131 | 123 | 113 |
| 131 | 120 | 108 |
| 127 | 117 | 107 |

$S_x = (a_2+ca_3+a_4)-(a_0+ca_7+a_6)$ ;  $S_x = 956$ ;  $S_y = (a_0+ca_1+a_2) - (a_6+ca_5+a_4)$ ;  $S_y = (199.1+2.61+61.1) - (60+-2.59+60.1)$ ;  $S_y = 744$

Tabel 4. Sumbu X

|            |     |     |     |     |     |     |     |     |
|------------|-----|-----|-----|-----|-----|-----|-----|-----|
| Sumbu<br>x | 956 | 915 | 891 | 907 | 931 | 954 | 968 | 970 |
|            | 930 | 892 | 870 | 882 | 906 | 933 | 959 | 968 |
|            | 897 | 872 | 858 | 862 | 884 | 912 | 947 | 962 |
|            | 884 | 867 | 860 | 849 | 867 | 894 | 933 | 953 |
|            | 913 | 884 | 879 | 843 | 856 | 882 | 921 | 944 |
|            | 935 | 897 | 890 | 849 | 856 | 876 | 906 | 927 |
|            | 910 | 884 | 880 | 863 | 867 | 874 | 888 | 903 |
|            | 880 | 869 | 874 | 873 | 879 | 877 | 882 | 893 |

Tabel. 5 Sumbu Y

|            |     |     |     |     |     |     |     |     |
|------------|-----|-----|-----|-----|-----|-----|-----|-----|
| Sumbu<br>Y | 744 | 685 | 661 | 679 | 705 | 714 | 716 | 730 |
|            | 714 | 664 | 648 | 666 | 684 | 695 | 709 | 732 |
|            | 691 | 656 | 640 | 650 | 664 | 676 | 697 | 730 |
|            | 676 | 667 | 642 | 643 | 649 | 660 | 689 | 727 |
|            | 683 | 688 | 651 | 635 | 638 | 648 | 679 | 720 |
|            | 689 | 669 | 642 | 639 | 646 | 654 | 672 | 697 |
|            | 714 | 688 | 648 | 639 | 647 | 652 | 668 | 693 |
|            | 666 | 649 | 642 | 659 | 665 | 657 | 652 | 665 |



G (Magnitudo) didapatkan menjadi:

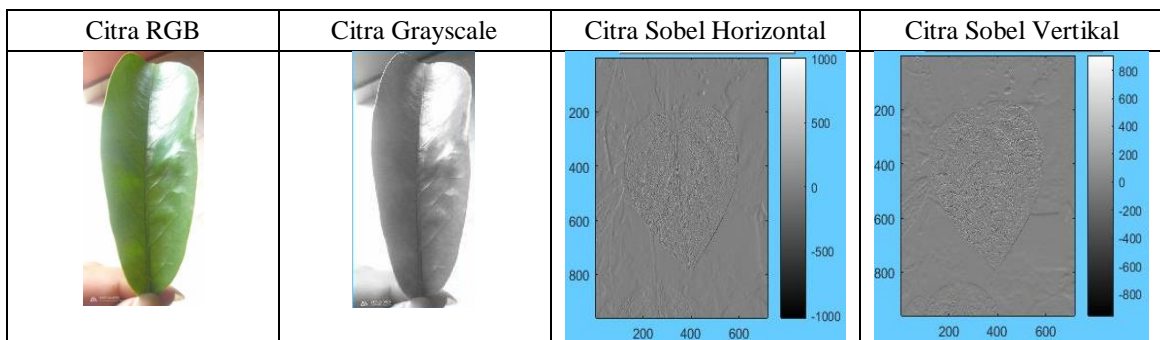
Tabel 6. Magnitudo

|          |          |          |          |          |          |          |          |
|----------|----------|----------|----------|----------|----------|----------|----------|
| 1211.393 | 1143     | 1109.415 | 1133     | 1167.812 | 1191.601 | 1204.027 | 1214.002 |
| 1172.474 | 1112.007 | 1084.806 | 1105.206 | 1135.206 | 1163.406 | 1192.628 | 1213.609 |
| 1132.294 | 1091.201 | 1070.404 | 1079.604 | 1105.6   | 1135.218 | 1175.848 | 1207.619 |
| 1112.849 | 1093.882 | 1073.203 | 1065.012 | 1083     | 1111.232 | 1159.832 | 1198.64  |
| 1140.201 | 1120.179 | 1093.82  | 1055.402 | 1067.605 | 1094.453 | 1144.239 | 1187.239 |
| 1161.441 | 1119.004 | 1097.39  | 1062.602 | 1072.405 | 1093.203 | 1128.016 | 1159.801 |
| 1156.675 | 1120.179 | 1092.842 | 1073.82  | 1081.803 | 1090.404 | 1111.201 | 1138.27  |
| 1103.61  | 1084.602 | 1084.454 | 1093.805 | 1102.21  | 1095.8   | 1096.826 | 1113.406 |

Jika  $G \geq 128$  maka nilai magnitudo menjadi 255, jika tidak nilai magnitudo menjadi 0:

Tabel 7. Hasil Metode Sobel

|     |     |     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 131 | 123 | 113 | 111 | 117 | 122 | 122 | 121 | 122 | 121 |
| 131 | 255 | 255 | 255 | 255 | 255 | 255 | 255 | 255 | 122 |
| 127 | 255 | 255 | 255 | 255 | 255 | 255 | 255 | 255 | 123 |
| 117 | 255 | 255 | 255 | 255 | 255 | 255 | 255 | 255 | 123 |
| 110 | 255 | 255 | 255 | 255 | 255 | 255 | 255 | 255 | 122 |
| 113 | 255 | 255 | 255 | 255 | 255 | 255 | 255 | 255 | 121 |
| 121 | 255 | 255 | 255 | 255 | 255 | 255 | 255 | 255 | 120 |
| 116 | 255 | 255 | 255 | 255 | 255 | 255 | 255 | 255 | 115 |
| 113 | 255 | 255 | 255 | 255 | 255 | 255 | 255 | 255 | 115 |
| 109 | 108 | 107 | 108 | 113 | 112 | 110 | 109 | 112 | 115 |



Gambar 2. Hasil Citra Metode Sobel

Metode prewitt menggunakan persamaan yang sama dengan operator sobel, hanya saja konstanta c yang digunakan bernilai 1. Operator prewitt tidak menekankan pembobotan pada piksel - piksel yang lebih dekat dengan titik pusat kernel. Metode ini mengambil prinsip dari fungsi laplacian yang dikenal sebagai fungsi untuk membangkitkan High Pass Filter. mengurangi sebelum melakukan perhitungan deteksi sehingga tepi-tepi yang dihasilkannya lebih banyak.

|       |   |   |    |       |    |    |    |
|-------|---|---|----|-------|----|----|----|
| $C_x$ | 1 | 0 | -1 | $C_y$ | -1 | -1 | -1 |
|       | 1 | 0 | -1 |       | 0  | 0  | 0  |
|       | 1 | 0 | -1 |       | 1  | 1  | 1  |

Gambar 3. Tabel Matriks Prewitt

$F(x,y)$  = ukuran piksel citra dalam matriks

$a_n$  = nilai piksel pada matriks

$(x,y)$  = nilai piksel baru hasil konvolusi

Turunan parsial dihitung dengan:

$$S_x = (a_2+a_3+a_4)-(a_0+a_7+a_6) = \text{Sumbu Vertikal}$$

$$S_y = (a_0+a_1+a_2)-(a_6+a_5+a_4) = \text{Sumbu Horizontal}$$

Tabel 8. Hasil Konversi

|     |     |     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 131 | 123 | 113 | 111 | 117 | 122 | 122 | 121 | 122 | 121 |
| 131 | 120 | 108 | 109 | 114 | 118 | 119 | 121 | 123 | 122 |
| 127 | 117 | 107 | 106 | 110 | 114 | 116 | 119 | 124 | 123 |
| 117 | 112 | 106 | 105 | 108 | 110 | 113 | 118 | 124 | 123 |
| 110 | 110 | 107 | 105 | 106 | 107 | 110 | 116 | 123 | 122 |
| 113 | 115 | 114 | 105 | 106 | 105 | 108 | 115 | 122 | 121 |
| 121 | 123 | 121 | 106 | 105 | 105 | 107 | 114 | 121 | 120 |
| 116 | 112 | 108 | 107 | 110 | 110 | 108 | 108 | 112 | 115 |
| 113 | 110 | 107 | 107 | 111 | 111 | 108 | 108 | 112 | 115 |
| 109 | 108 | 107 | 108 | 113 | 112 | 110 | 109 | 112 | 115 |

Proses iterasi sebanyak  $n$ :

Tabel 9. Hasil Iterasi Sumbu x Prewitt

|            |     |     |     |     |     |     |     |     |
|------------|-----|-----|-----|-----|-----|-----|-----|-----|
| Sumbu<br>X | 848 | 806 | 777 | 789 | 812 | 833 | 845 | 848 |
|            | 823 | 786 | 760 | 768 | 790 | 814 | 835 | 845 |
|            | 791 | 767 | 750 | 752 | 771 | 794 | 823 | 839 |
|            | 777 | 762 | 754 | 742 | 757 | 778 | 810 | 831 |
|            | 799 | 779 | 773 | 738 | 748 | 767 | 799 | 823 |
|            | 814 | 791 | 785 | 744 | 749 | 762 | 785 | 807 |
|            | 802 | 777 | 770 | 753 | 759 | 766 | 776 | 788 |
|            | 773 | 762 | 763 | 762 | 771 | 769 | 770 | 778 |

Tabel 10. Iterasi Sumbu y Prewitt

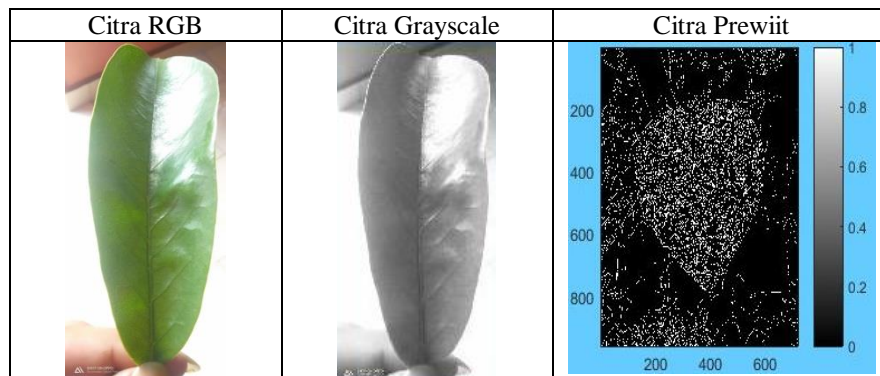
|         |      |      |      |      |      |      |      |      |
|---------|------|------|------|------|------|------|------|------|
| sumbu y | -958 | -897 | -881 | -907 | -937 | -952 | -964 | -976 |
|         | -926 | -874 | -864 | -886 | -910 | -931 | -957 | -978 |
|         | -905 | -866 | -852 | -864 | -884 | -908 | -943 | -974 |
|         | -904 | -877 | -854 | -853 | -865 | -890 | -933 | -969 |
|         | -925 | -900 | -861 | -845 | -852 | -876 | -921 | -960 |
|         | -905 | -883 | -862 | -859 | -862 | -870 | -896 | -927 |
|         | -928 | -902 | -870 | -861 | -863 | -868 | -892 | -923 |
|         | -880 | -865 | -868 | -883 | -885 | -875 | -876 | -895 |

Tabel 11. Hasil Magnitudo

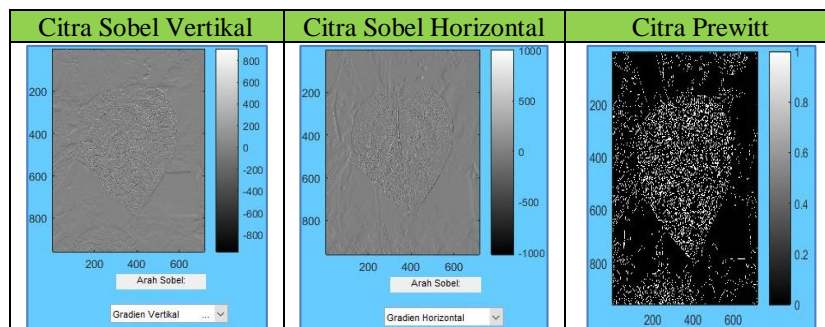
|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| -110 | -91  | -104 | -118 | -125 | -119 | -119 | -128 |
| -103 | -88  | -104 | -118 | -120 | -117 | -122 | -133 |
| -114 | -99  | -102 | -112 | -113 | -114 | -120 | -135 |
| -127 | -115 | -100 | -111 | -108 | -112 | -123 | -138 |
| -126 | -121 | -88  | -107 | -104 | -109 | -122 | -137 |
| -91  | -92  | -77  | -115 | -113 | -108 | -111 | -120 |
| -126 | -125 | -100 | -108 | -104 | -102 | -116 | -135 |
| -107 | -103 | -105 | -121 | -114 | -106 | -106 | -117 |

**Tabel 12. Grayscale dan Magnitudo**

|     |     |     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 131 | 123 | 113 | 111 | 117 | 122 | 122 | 121 | 122 | 121 |
| 131 | 120 | 108 | 109 | 114 | 118 | 119 | 121 | 123 | 122 |
| 127 | 117 | 107 | 106 | 110 | 114 | 116 | 119 | 124 | 123 |
| 117 | 112 | 106 | 105 | 108 | 110 | 113 | 118 | 124 | 123 |
| 110 | 110 | 107 | 105 | 106 | 107 | 110 | 116 | 123 | 122 |
| 113 | 115 | 114 | 105 | 106 | 105 | 108 | 115 | 122 | 121 |
| 121 | 123 | 121 | 106 | 105 | 105 | 107 | 114 | 121 | 120 |
| 116 | 112 | 108 | 107 | 110 | 110 | 108 | 108 | 112 | 115 |
| 113 | 110 | 107 | 107 | 111 | 111 | 108 | 108 | 112 | 115 |
| 109 | 108 | 107 | 108 | 113 | 112 | 110 | 109 | 112 | 115 |

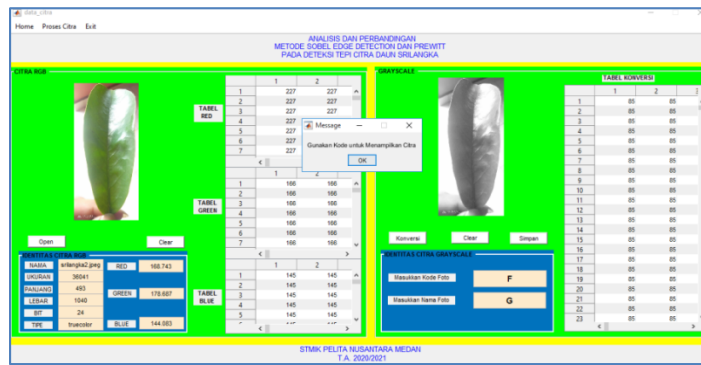


Gambar 4. Citra Hasil Prewitt



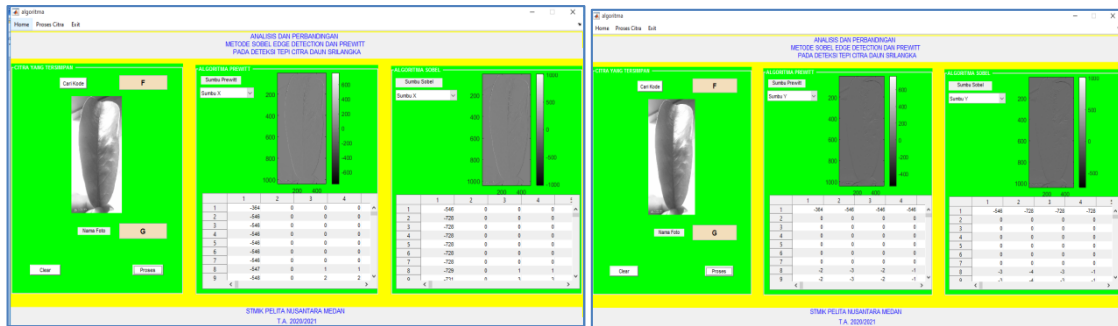
Gambar 5. Histogram Sobel dan Prewitt

Setelah ekstrak kedalam nilai Red, Green dan Blue, maka selanjutnya di konversikan ke citra grayscale, nilai Red, Green, dan Blue ditambahkan kemudian dibagi dengan angka 3. Didalam panel grayscale terdapat tombol konversi yang digunakan untuk mengkonversi citra, tombol clear yaitu untuk menghapus atau mengambil file citra lain, dan tombol save untuk menyimpan citra. Cara menyimpan citra grayscale yaitu dengan mengisikan terlebih dahulu kode foto dan nama foto. Kode dan nama foto dibuat sesuai dengan keinginan user. Kode dan nama foto yang diisi akan disimpan ke dalam database.

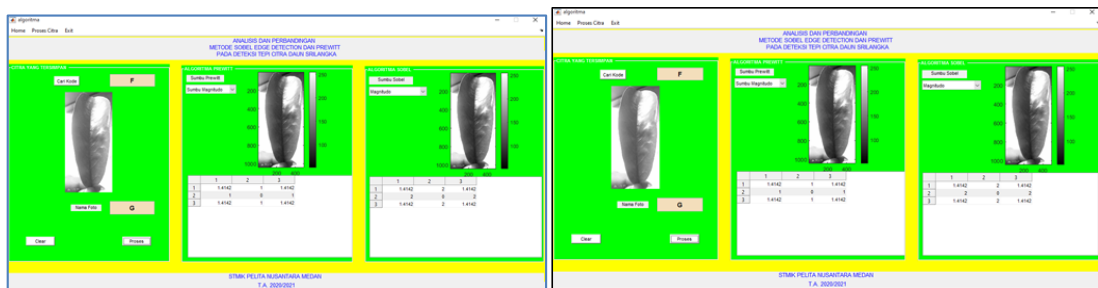


Gambar 6. Menampilkan Inputan Citra

Ketika mengetik kode foto yang telah dimasukkan, selanjutnya mengklik tombol cari foto maka akan menampilkan foto yang telah tersimpan. Tombol clear untuk menghapus atau membatalkan citra yang dipilih. Tombol Proses digunakan untuk menampilkan proses.



Gambar 7. Bangkitkan Sumbu x dan y Sobel dan Prewitt



Gambar 8. Bangkitkan Magnitudo Sobel dan Prewitt

## 5. Kesimpulan

Kesimpulan dari penelitian :

1. Sobel dan Prewitt mampu mengidentifikasi tepi dengan sumbu-sumbu yang digunakan.
2. Sobel dan Prewitt memiliki intensitas bilangan acak yang hasilnya berupa angka 0 dan 255 secara menyebar.
3. Sobel dan Prewitt digunakan untuk menganalisis tepi citra daun srilanka dan histogram ditampilkan untuk mengukur intensitas hasil metode sobel dan prewitt dalam bentuk grafik

## 6. Daftar Pustaka

- [1] O. F. B. Barus, "Penerapan Metode Robert Pada Deteksi Tepi Citra Split Underwater," J. Media Inform. Budidarma, vol. 2, no. 1, pp. 35–44, 2018, doi: 10.30865/mib.v2i1.995.
- [2] A. S. Sinaga and P. Marpaung, "EKSTRAK CIRI KOMUNIKASI NONVERBAL MENGGUNAKAN GRAY LEVEL CO-OCCURRENCE MATRIX," vol. 12, no. 2, pp. 43–49, 2020.



- [3] K. Letelay, "Perbandingan Kinerja Metode Deteksi Tepi," *J-Icon*, vol. 7, no. 1, pp. 1–8, 2019.
- [4] A. S. R. Maryana, "Real Time Database Seleksi Wajah Digital Menggunakan Algoritma CAMshfit Abstrak," *Fountainins Infomatics J. Unida Gontor*, vol. 5, no. 1, pp. 9–13, 2020.
- [5] M. M. Sobel, R. Canny, P. Teguh, K. Putra, N. Kadek, and A. Wirdiani, "Pengolahan Citra Digital Deteksi Tepi Untuk Membandingkan Metode Sobel, Robert dan Canny," *J. Ilm. Merpati (Menara Penelit. Akad. Teknol. Informasi)*, vol. 2, no. 2, pp. 253–261, 2016.
- [6] W. Supriyatin, "Perbandingan Metode Sobel, Prewitt, Robert dan Canny pada Deteksi Tepi Objek Bergerak," *Ilk. J. Ilm.*, vol. 12, no. 2, pp. 112–120, 2020, doi: 10.33096/ilkom.v12i2.541.112-120.
- [7] I. G. N. Suryantara, "Implementasi Deteksi Tepi Untuk Mendeteksi Keretakan Tulang Orang Lanjut Usia (Manula) Pada Citra Rontgen Dengan Operator Sobel Dan Prewitt," *J. Algoritm. Log. dan Komputasi*, vol. 1, no. 2, pp. 51–60, 2018.
- [8] Q. Xu, S. Varadarajan, C. Chakrabarti, and L. J. Karam, "A distributed canny edge detector: Algorithm and FPGA implementation," *IEEE Transactions on Image Processing*, vol. 23, no. 7, pp. 2944–2960, 2014.
- [9] E. Winarno, "Aplikasi Deteksi Tepi pada Realtime Video Menggunakan Algoritma Canny Detection," *None*, vol. 16, no. 1, p. 247202, 2011.
- [10] R. Wahyusari, "Pemanfaatan Mathematical Morphology Untuk Deteksi Tepi Batik," *Simetris J. Tek. Mesin, Elektro dan Ilmu Komput.*, vol. 8, no. 1, pp. 389–392, 2017, doi: 10.24176/simet.v8i1.1027.
- [11] A. Pranata and E. Z. Astuti, "Pengolahan Citra Berbasis Deteksi Tepi Prewitt Pada Gambar Gigi Manusia," *Eksplora Inform.*, vol. 6, no. 2, pp. 98–105, 2017, [Online]. Available: <https://eksplora.stikom-bali.ac.id/index.php/eksplora/article/view/110/91>.
- [12] Y. T. Rada, "Pengenalan Pola Pada Fisik Mobil Menggunakan Persamaan Diferensial Deteksi Tepi (Edge Detection)," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 7, no. 1, p. 57, 2018.
- [13] E. R. Kuswandi and N. Fadillah, "Perbandingan Metode Robert dan Metode Prewitt untuk Deteksi Tepi pada Citra Tanda Tangan," *InfoTekJar (Jurnal Nas. Inform. dan Teknol. Jaringan)*, vol. 3, no. 2, pp. 155–158, 2019.
- [14] F. Arnia and R. Muharar, "Deteksi Pemalsuan Citra dengan Teknik Copy-Move Menggunakan Metode Ordinal Measure dari Koefisien Discrete Cosine Transform," *J. Nas. Tek. Elektro*, vol. 5, no. 2, pp. 165–174, 2016.